Parameter-less, Population-sizing DSMGA-II

Marcin M. Komarnicki
Department of Computational Intelligence
Wroclaw University of Science and Technology
Wroclaw, Poland
marcin.komarnicki@pwr.edu.pl

Michal W. Przewozniczek
Department of Computational Intelligence
Wroclaw University of Science and Technology
Wroclaw, Poland
michal.przewozniczek@pwr.edu.pl

ABSTRACT

Limiting the number of required settings is an important part of any evolutionary method development. The final objective of this process is a method version that is parameter-less. Based on the research results presented that far, the leading methods in the combinatorial optimization are Linkage Tree Genetic Algorithm (LTGA), Parameter-less Population Pyramid (P3) and Dependency Structure Matrix Genetic Algorithm II (DSMGA-II). P3 was originally proposed as a parameter-less method, while LTGA and DSMGA-II in their original propositions both require one parameter that is the population size. Recently, a population-sizing technique was used to propose a parameter-less version of LTGA (psLTGA). However, the population-sizing was not introduced in DSMGA-II that misses its effective parameter-less version. Therefore, to fill this gap, in this paper we propose a Population-sizing DSMGA-II (psDSMGA-II) that is parameter-less. We also show that psDSMGA-II is more effective than its predecessor and that it may successfully compete with psLTGA and P3.

CCS CONCEPTS

• **Computing methodologies → Search methodologies**;

KEYWORDS

Population-sizing, Linkage Learning, Genetic Algorithms, Dependency Structure Matrix

ACM Reference Format:
Marcin M. Komarnicki and Michal W. Przewozniczek. 2019. Parameter-less, Population-sizing DSMGA-II. In *Proceedings of Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3322080

1 INTRODUCTION

The configuration of evolutionary method may significantly influence the final results quality. Therefore, an important step to do reliable research in the field of Evolutionary Computation is careful tuning. In this paper, we show that even properly made tuning may not lead to good quality results. If the considered test case set contains test cases that demand different settings then the universal configuration may not exist. Linkage Tree Genetic Algorithm (LTGA) [7] is one of the up-to-date evolutionary methods. Recently, the populations-sizing technique was introduced into it, making it parameter-less [1]. Another effective, up-to-date evolutionary method is the Dependency Structure Matrix Genetic Algorithm (DSMGA-II) [3]. DSMGA-II was shown effective in solving theoretical and practical problems [3, 6] but misses an parameter-less version. Therefore, to fill this gap, we adjust and introduce the population-sizing technique into DSMGA-II, making it a parameter-less and self-adapting method.

Making any method parameter-less is always comfortable for the user. Moreover, the self-adjustment mechanisms of the parameter-less method may make it more effective. The example of such a method is Population-Sizing Linkage Tree Genetic Algorithm (psLTGA) [1] which introduces the population-sizing into LTGA making it parameter-less. psLTGA maintains multiple LTGA instances, each with the different population size. Starting with LTGA containing only one individual, the iteration of LTGA with the doubled population size is executed at each 4^{th} iteration. During the psLTGA run, useless LTGA populations are dropped. Single LTGA population is found useless if all of its individuals are the same or its average population fitness is worse than the average fitness of at least one LTGA with the larger population size. Additionally, LTGA populations of a size smaller than those already dropped are found useless as well and also dropped. All LTGA populations are mainly independent, only during the FI phase the best individual found so far by any LTGA population is used as the donor individual.

2 POPULATION-SIZING DSMGA-II

The proposed psDSMGA-II uses the same population-sizing scheme as psLTGA. psDSMGA-II introduces the following cooperation mechanism between different DSMGA-II populations. When a DSMGA-II population successfully uses the restricted mixing on any of its individuals, then the back mixing is executed, and positive change from one individual is injected into other individuals and preserved if it does not decrease the fitness (otherwise it is reverted). The globally best individual is also affected by this back mixing operation. Additionally, for any DSMGA-II population, the restricted mixing operation is also executed for the globally best individual. However, even if restricted mixing on the globally best individual is successful, no back mixing operation is performed. In addition, we also consider a version without any cooperation between DSMGA-II populations. This method is denoted as psDSMGA-II-Simple.

3 THE RESULTS

For the experiments, we use test problems typical for the field of Genetic Algorithms [2–4, 7]. All methods were coded in C++ and joined in one project (https://github.com/kommar/psDSMGA-II). The source codes repository also contains settings files, detailed

---

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322080

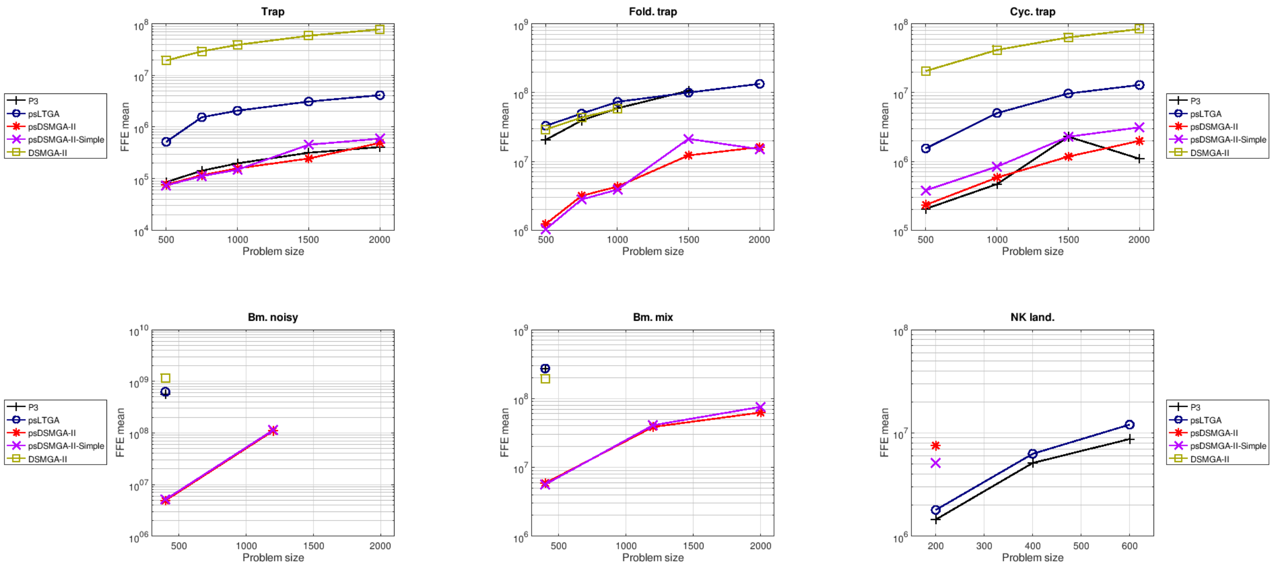

Figure 1: FFE scalability of considered methods

results of all runs, and results summary. For P3 and all DSMGA-II versions we use the source code given in [2] and [3] respectively. For psLTGA we use the source code published by its Authors[1].

To perform comparisons, we need to define the population size for DSMGA-II. For this purpose we use the results proposed by psDSMGA-II. The population containing 2^{15} individuals was large enough to find an optimum in over 99% of the runs. Therefore, it was employed as DSMGA-II population size. We use a time-based stop condition, since P3 and DSMGA-II use fitness caching and psLTGA does not. Thus, the stop condition based on the number of fitness functions evaluations (FFE) may be unfair [5].

As presented in Fig. 1, psDSMGA-II scales better than DSMGA-II for all considered problems. For Trap functions psDSMGA-II and P3 scale significantly better than other methods. For Folded trap, psDSMGA-II outperforms all other methods. For Cycling traps psDSMGA-II scales similarly to P3 and outperform psLTGA. Finally, for Bimodal Noised and Bimodal Mixed concatenations psDSMGA-II significantly outperform other methods. On the other hand, P3 and psLTGA are significantly better for NK-landscapes.

There are no significant differences in scalability and results quality between psDSMGA-II and psDSMGA-II-Simple. The conclusion is that the repeated executions of DSMGA-II with different population sizes have a decisive influence on the method effectiveness. The repeated executions with the increased population sizes seem to be enough to obtain both: the increase of results quality and the self-adaptation.

4 CONCLUSION

We have proposed a parameter-less DSMGA-II by introducing a population-sizing technique to it. psDSMGA-II outperformed well-tuned DSMGA-II. Obtained results indicate that psDSMGA-II is highly competitive with effective, up-to-date parameter-less methods like P3 and psLTGA. The main future work direction is to use the proposed psDSMGA-II in real-world problems optimization. For instance, such application may lead to the improvement of DSMGA-II results quality presented in [6].

ACKNOWLEDGMENTS

This work was supported by the Polish National Science Centre (NCN) under Grant 2015/19/D/ST6/03115 and the statutory funds of the Department of Computational Intelligence.

REFERENCES

[1] Peter A.N. Bosman, Ngoc Hoang Luong, and Dirk Thierens. 2016. Expanding from Discrete Cartesian to Permutation Gene-pool Optimal Mixing Evolutionary Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016 (GECCO '16)*. ACM, 637–644.

[2] Brian W. Goldman and William F. Punch. 2014. Parameter-less Population Pyramid. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. 785–792.

[3] Shih-Huan Hsu and Tian-Li Yu. 2015. Optimization by Pairwise Linkage Detection, Incremental Linkage Set, and Restricted / Back Mixing: DSMGA-II. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. 519–526.

[4] Marcin M. Komarnicki and Michal W. Przewozniczek. 2017. Parameter-less Population Pyramid with Feedback. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '17)*. ACM, New York, NY, USA, 109–110. https://doi.org/10.1145/3067695.3076065

[5] Michal W. Przewozniczek and Marcin M. Komarnicki. 2018. The Influence of Fitness Caching on Modern Evolutionary Methods and Fair Computation Load Measurement. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '18)*. ACM, New York, NY, USA, 241–242. https://doi.org/10.1145/3205651.3205788

[6] Michal Witold Przewozniczek, Krzysztof Walkowiak, Arunabha Sen, Marcin Komarnicki, and Piotr Lechowicz. 2019. The transformation of the k-Shortest Steiner trees search problem into binary dynamic problem for effective evolutionary methods application. *Information Sciences* 479 (2019), 1 – 19. https://doi.org/10.1016/j.ins.2018.11.015

[7] Dirk Thierens and Peter A.N. Bosman. 2013. Hierarchical problem solving with the linkage tree genetic algorithm. In *Proceeding of the 2013 Annual Conference on Genetic and Evolutionary Computation Conference*. ACM, 877–884.

---

[1] https://homepages.cwi.nl/~bosman/source_code.php